(2)

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	30 October 1990	Final, 1 April 1988 - 30 June 1990

4. TITLE AND SUBTITLE

Large Sparse Stable Matrix Computations

5. FUNDING NUMBERS

AFOSR-88-0161

6. AUTHOR(S)

Alex Pothen and Jesse L. Barlow

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Department of Computer Science
The Pennsylvania State University
University Park, PA 16802

8. PERFORMING ORGANIZATION REPORT NUMBER

AFOSR-TR- 90 1156

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

AFOSR/NM
Bldg. 410
Bolling AFB
Washington, D.C. 20332-6448

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

AD-A229 837

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

The project proposal discussed two problem areas;
(1) The solution of large sparse systems of linear equations; AMD
(2) The solution of sparse least squares problems.
We report significant progress in both of these areas and in a third area, the solution of the algebraic eigenvalue problem.

The progress in solving systems of linear equations included an algorithm for computing ordering for efficiently factoring sparse symmetric, positive definite systems in parallel. We also made important progress in computing the ordering itself in parallel. Other progress included a method for handling singular blocks in a one-way dissection ordering and an error analysis of Gaussian elimination in unnormalized arithmetic.

(See attached sheet) OVER

14. SUBJECT TERMS

Sparse matrix computations, linear equations, least squares problems, eigenvalue problems, graph algorithms.

15. NUMBER OF PAGES

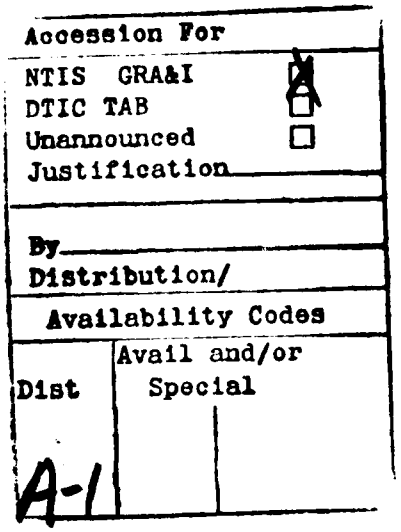
7

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

For linear least squares problems we developed an efficient reliable method for detecting the rank of a sparse matrix without column exchanges. The method used a static data structure. We also analyzed and compared methods for computing sparse and dense QR factorizations on message passing architectures.

On the algebraic eigenvalue problem, we participated in resolving long standing open questions on relative perturbation bounds on certain diagonally dominant eigenvalue problems.



Final Report AFOSR-88-0161
Large, Sparse, Stable Matrix Computations
4/1/88 — 6/30/90

Alex Pothen

This proposal considered two major problems in sparse matrix computations: the solution of sparse systems of linear equations and the solution of unconstrained and constrained least squares problems.

1. Sparse systems of linear equations. We have designed algorithms for computing orderings for efficiently factoring sparse, symmetric positive definite matrices in parallel. Jess and Kees (1982) had suggested a two-step approach for computing a good parallel ordering: (i). compute a fill-reducing ordering of the matrix; (ii). reorder the filled chordal graph of the Cholesky factor L from the step (i) to obtain an ordering that permits the maximum possible parallelism at each step, without incurring any additional fill. The ordering obtained from step (ii) leads to the shortest elimination tree of the Cholesky factor L which preserves the fill in L

Previous algorithms for implementing step (ii) had space and time requirements that were much greater than the requirements of step (i); hence these algorithms were impractical for large problems. In [1], in joint work with Lewis and Peyton, we designed an algorithm for the second step that was linear in the number of compressed subscripts for L by making use of a new data structure, the clique tree. This algorithm requires much less space and time than the initial fill-reducing ordering. We also provided some justification for this two-step approach by showing that the problem of computing the parallel ordering of A that leads to the shortest elimination tree irrespective of the fill is NP-complete and hence intractable [3].

We have considered an algebraic approach to computing good orderings in parallel. The paper [12] (joint work with Simon) concerns the design of a parallel algorithm for computing the parallel ordering. It presents a solution to the problem of computing good separators (this is one step in a parallel ordering algorithm), which makes use of the eigenvectors of the Laplacian matrix of a graph. This spectral approach computes smaller separators than the Automated Nested Dissection algorithm. We have also shown that lower bounds on separator sizes can be obtained from the eigenvalues of the Laplacian matrix. A third paper (in preparation) [11] makes use of the above algebraic approach for a parallel algorithm for reducing the envelope of sparse matrices.

We have also investigated the role played by the clique tree data structure (introduced to compute a Jess and Kees ordering) in other sparse matrix problems.

We have completed an implementation [14] of a multifrontal sparse Cholesky factorization algorithm for an iPSC/2 hypercube, and are involved in studying the influence of good orderings, mappings, and clique tree structures on its efficiency. In other work [2], we have been studying what constitutes a 'supernode', a group of columns of the Cholesky factor L that forms a maximal dense submatrix. The difficulty is that this concept depends on the algorithmic context. It can be shown that (according to one definition) a set of supernodes can be obtained from the clique tree by an $O(n)$ time algorithm. (Here n is the order of the matrix.) Supernodes have been used by Ashcraft et al. (1987) to enable vectorization in the computation of sparse numerical factorization at speeds comparable to dense matrix factorizations on vector supercomputers. The report [4] describes our preliminary study of supernodes in sparse factorizations.

The clique tree is a compact representation of the structure of the Cholesky factor L. Nevertheless, at the expense of some computation, it is possible to obtain more compact representations. In [13], we have investigated a data structure called the compact clique tree. This data structure has applications in communication efficient parallel factorization algorithms and in storage efficient

out-of-core algorithms. We have proved that the compact clique tree never requires storage greater than the skeleton graph (a concept introduced by Liu), and on some worst-case examples requires only $\Theta(n)$ storage when the skeleton graph requires $\Theta(n^2)$ storage. On all of the computational problems that we have experimented with, the two require almost identical storage; this is typically about a fourth of the storage needed for the clique tree.

2. Linear least squares problems. Work has been done on several issues that arise in the solution of dense and sparse least squares problems on both sequential and parallel computers.

An algorithm for computing the block upper triangular form of rectangular and square sparse matrices was described and implemented in [6]. This form computes the irreducible blocks of the given matrix, and since only these blocks need to be factored to solve linear systems and least squares problems, can save the storage and time needed to factor the given matrix. It is also is useful in correctly computing the nonzero structures of the factor matrices Q and R during symbolic factorization.

The paper [5] presents an algorithm for computing a sparse basis for the null space of the equilibrium matrix of a physical structure. This problem arises in the solution of equality constrained least squares problems in structural analysis. By making use of the additional information available in the physical situation, sparser null bases could be computed; the algorithm was also faster than previous algorithms.

Several parallel algorithms for computing the orthogonal factorization of a dense matrix on a distributed memory multiprocessor were described in [7,8,9]. Both Givens and Householder orthogonalization algorithms were considered. The time and communication complexities of the algorithms were analyzed, and shown to agree with the time taken by the algorithms.

The above work laid the foundation for our work on the sparse linear least squares problem. The paper [10] describes a parallel algorithm for the numerical factorization step (the dominant step in the time complexity) in the orthogonal decomposition of a sparse matrix on a hypercube multiprocessor. This algorithm computes the orthogonal factorization by means of a sequence of submatrix merges involving upper trapezoidal matrices. The merges are performed by the use of row oriented Householder transformations, and is organized around a merge tree data structure. We have shown that this algorithm has small arithmetic and communication complexities and we obtain good parallel efficiencies in our implementation on an iPSC/2 hypercube multiprocessor.

REFERENCES

[1] J. G. Lewis, B. W. Peyton, and A. Pothen. A fast algorithm for reordering sparse matrices for parallel factorization. *SIAM J. Sci. Stat. Comput.*, 10:1146–1173, 1989. (Special section on Sparse Matrix Algorithms for Supercomputers).

[2] J. G. Lewis, B. W. Peyton, and A. Pothen. Clique trees and supernodes in in sparse factorizations. Work in preparation, Sep 1990.

[3] A. Pothen. The complexity of optimal elimination trees. Technical Report CS-88-16, Computer Science, Pennsylvania State University, University Park, PA, April 1988.

[4] A. Pothen. Simplicial cliques, shortest elimination trees, and supernodes in sparse cholesky factorization. Technical Report CS-88-13, Computer Science, The Pennsylvania State University, University Park, PA, April 1988.

[5] A. Pothen. Sparse null basis computations in structural optimization. *Numer. Math.*, 55:501–519, 1989.

[6] A. Pothen and C-J. Fan. Computing the block triangular form of a sparse matrix. To appear, ACM Trans. Math. Softw., 16 (4), Dec 1990.

[7] A. Pothen, S. Jha, and U. Vemulapati. Orthogonal factorization on a distributed memory multiprocessor. In M. T. Heath, editor, *Hypercube Multiprocessors 1987*, pages 587–596. SIAM, Philadelphia, 1987.

[8] A. Pothen and P. Raghavan. Distributed orthogonal factorization. In G. Fox, editor, *Proceedings of the Third Conference on Hypercube Concurrent Computers and Applications*, pages 1610–1620. ACM, 1988.

[9] A. Pothen and P. Raghavan. Distributed orthogonal factorization: Givens and Householder algorithms. *SIAM J. Sci. Stat. Comput.*, 10:1113–1134, Nov 1989.

[10] A. Pothen and P. Raghavan. Sparse orthogonal factorization on a *distributed memory* multiprocessor. submitted, Sep 1990.

[11] A. Pothen and H. D. Simon. A parallel iterative algorithm for envelope reduction in sparse matrices. in preparation, 1990.

[12] A. Pothen, H. D. Simon, and K. P. Liou. Partitioning sparse matrices with eigenvectors of graphs. *SIAM J. Mat. Anal. Appl.*, 11:430–452, 1990. Special section containing papers from the SIAM Conference on Sparse Matrices.

[13] A. Pothen and C. Sun. Compact clique tree data structures in sparse matrix factorizations. In *Large-Scale Numerical Optimization, Proceedings in Applied Mathematics, (Proceedings of a Workshop held at the Mathematical Sciences Institute, Cornell University)*. Society for Industrial and Applied Mathematics, Nov 1990.

[14] A. Pothen and C. Sun. An efficient distributed multifrontal method using clique trees. in preparation, 1990.

Final Report AFOSR-88-0161
Large Sparse Stable Matrix Computations
4/1/88 - 6/30/90

Jesse L. Barlow

The project proposal discussed two problems in sparse matrix computations:(1) the solution of systems of linear equations ; (2) the solution of linear least squares problems. We made progress in both of these areas and in a third area: the solution of the algebraic eigenvalue problem. The work in linear least squares resulted in a new direction not discussed in the proposal. Our results have implications in the development of stable parallel algorithms to solve these problems.

1. The Solution of Linear Systems of Equations. This part of the effort yielded two significant results. The first concerned the technique of one-way dissection or domain decomposition. It is commonly used in the numerical solution of partial differential equations. The resulting system has a block bordered form which is nearly ideal for parallel computation.

In some contexts, in particular, in the solution of Stokes flow problems [22] , the diagonal blocks may be singular even though the matrix is nonsingular. We developed an improvement of a direct method for solving this problem due to Gunzberger and Nicholaides [22]. We examined the stability of various methods for resolving the singularity and proposed a better back substitution algorithm [8]. We tested our algorithm on the Intel iPSC/1 and it produced linear speedups.

Special purpose devices have an important role in sparse matrix computations. Digit online arithmetic is often used for special purpose devices because its ability to "pipe on digits" speeds up computations by factors ranging from 2 to 16 [20]. A PhD thesis from Penn State [29] pointed out that such arithmetics produce unnormalized results and thus numerical algorithms may not have the same stability properties. Cavellaro et al.[15] recently used this arithmetic in the design of a robot control device.

As an example, we considered the error analysis of Gaussian elimination in unnormalized arithmetic [12]. The algorithm exhibits exactly the same properties as in standard arithmetic for the diagonally dominant matrices that arise in the numerical solution of partial differential equations. For general matrices, there are subtle differences between the error analysis for unnormalized arithmetic and standard floating point arithmetic.

2. The Solution of Linear Least Squares Problems. We developed algorithms for two different constrained least squares problems. We have also developed an efficient method for detecting the rank of a sparse matrix. The method allows one to use a static data structure throughout the algorithm. Finally, we proved the stability of a parallel method for computing the sample variance.

For the constrained least squares problem

$$\min_{x \in \Re^n} \| f - Ex \|_2 \tag{1}$$

subject to the constraints

$$Cx = g \tag{2}$$

where C is an $m_1 \times n$ matrix, E is an $m_2 \times n$ matrix, we made progress on direct approaches. The error analysis of a weighting procedure with deferred correction

was completed [5,1]. A sparse version of the algorithm was developed in [10,11,28]. There were two important results from this. The first is that the deferred correction procedure converges in only two iterations for a large class of problems [11]. This development also lead to a more robust implementation of the strategy. The second development concerned the the problem of accurately detecting the rank of C [10] which is crucial to the solution of (1)-(2). We developed a strategy which is about as accurate as maximal column pivoting and is provably more accurate than the strategy in SPARSPAK-B [21]. The strategy allows one to use a static data structure. It can be used to detect the rank of a general sparse matrix and does not require access to the elimination tree as does a related approach by Bischof, Pierce, and Lewis [13]. This rank detection procedure can also be used in conjunction with the iterative procedure for (1)-(2) given by Barlow, Nichols, and Plemmons [6].

Joint work with G. Toraldo [7] is in progress on the solution of the bound constrained quadratic programming problem

$$\min_{u \in \Re^n} \frac{1}{2} u^T A u - u^T b \tag{3}$$

subject to

$$c \leq x \leq d \tag{4}$$

where c and d are n-vectors. We have considerd the projected gradient strategy for this problem due to More' and Toraldo [26,27]. This strategy tends to find the active set much faster than classical active set strategies. We show that if A can be scaled to the form

$$A = I - N$$

where N is symmetric, positive definite, then the strategy will always take large projected gradient steps. The results can be strengthened considerably if A is a Steiltjes matrix, and the constraint (4) is

$$u \geq 0. \tag{5}$$

In that case, if A is positive definite, we are solving the linear complementary problem. Our simplifications to the projected gradient strategy avoid evaluations of $F(u) = \frac{1}{2} u^T A u - u^T b$ entirely. Our strategy solved a free boundary problem with 3337 variables in 85.62 seconds on a SUN/4. The underlying linear equation solver was Jacobi preconditioned conjugate gradient. However, we think that the method may be suitable for use with direct factorization methods for A. Preliminary results are in the report [7].

An incidental result was the error analysis of an algorithm due to Chan, Golub, and LeVeque [16] for computing the sample variance. The algorithm computes the variance of a sample of size n in $O(\log n)$ parallel steps with only one pass through the sample data. Its stability had been an open question. Our analysis showed that the algorithm was indeed stable.

3. The Solution of the Algebraic Eigenvalue Problem. The algebraic eigenvalue problem is that of solving

$$Ax = \lambda Bx \tag{6}$$

for the scalar λ and the n-vector x. In our research, we considered the case where A and B are symmetric $n \times n$ matrices and B is positive definite. The problem (6) arises in structural analysis.

In [4], we considered the problem where A is of the form

$$A = \Delta(E + N)\Delta. \tag{7}$$

Here Δ and E are diagonal matrices, Δ is nonsingular, $\| N \|_2 = \gamma < 1$, and the diagonals of E are ± 1. Thus A is diagonally dominant in the Euclidean matrix norm. This class of matrices includes all consistently ordered diagonally dominant matrices that are solvable by the classical iterative methods.

We show that for the case where $B = I$ or A is positive definite, we can obtain much better perturbation bounds on both the eigenvalues and eigenvectors than presently given in the literature. These results resolved a well known open question regarded relative error bounds on eigenvalues.

These bounds led to more robust algorithms that will be incorporated in the LAPACK linear algebra library [4] for shared memory high performance architectures. Some of the algorithmic questions arising from this analysis have been answered. So far, only bisection followed by inverse iteration has been shown to satisfy all of the bounds discussed in [4]. For the singular value decomposition, the Jacobi algorithm [18] achieves these bounds and the QR algorithm achieves them for the bidiagonal SVD. However, the QR for general symmetric tridiagonal matrices does not achieve relative accuracy. For the new divide-and-conquer algorithms that have been considered for implementation in LAPACK, no results on relative accuracy have been proven or disproven.

In [24,25,23], we develop bounds on the eigenvalues of banded Toeplitz matrices. We also develop a method for finding the eigenvalues of these matrices that uses the Bunch, Nielsen, and Sorensen [14] update procedure. For banded symmetric Toeplitz matrices, the procedure is faster than the QR algorithm and just as stable.

REFERENCES

[1] J.L. Barlow, Error Analysis and Implementation Aspects of Deferred Correction for Equality Constrained Least Squares Problems, SIAM J. Num. Anal. 25,(1988),pp.892-906.

[2] J.L. Barlow, Error Analysis of a Pairwise Summation Algorithm to Compute the Sample Variance, Tech. Rep. CS-90-10, Department of Computer Science, The Pennsylvania State University, University Park, PA, 1990, to appear, Numerische Mathematik.

[3] J.L. Barlow, On the Use of Structural Zeros in Orthogonal Factorization, SIAM J. Sci. Stat. Computing 11,(1990), pp.600-601.

[4] J.L. Barlow and J.W. Demmel, Computing Accurate Eigensystems of Scaled Diagonally Dominant Matrices, SIAM J. Num. Anal. 27,(1990),pp.762-791.

[5] J.L. Barlow and S.L. Handy, The direct solution of weighted and equality constrained least squares problems, SIAM J. Sci. Stat. Computing 9,(1988),pp.1340-1358.

[6] J.L. Barlow, N.K. Nichols, and R.J. Plemmons, Iterative Methods for Equality-Constrained Least Squares Problems, SIAM J. Sci. Stat. Computing 9,(1988), pp.892-906.

[7] J.L. Barlow and G. Toraldo, Projected Gradient Methods for the Solution of the M-matrix Linear Complementary Problem (Working Paper), Tech. Rep. CS-90-43, Department of Computer Science, The Pennsylvania State University, University Park, PA, 1990.

[8] J.L. Barlow and U.B. Vemulapati, An Improved Method for One-Way Dissection with Singular Diagonal Blocks, Tech. Rep. CS-89-05, Department of Computer Science, The Pennsylvania State University, University Park, PA, April 1988(revised January 1989), to appear, SIAM J. Matrix Anal. Appl., October 1990.

[9] J.L. Barlow and U.B. Vemulapati, Incremental Condition Estimation for Parallel Sparse Matrix Factorizations, Proc, Fifth Distributed Memory Computing Conference, D. Walker and Q. Stout, Eds., IEEE Computer Society Press, Los Alamitos, CA, 1990, pp.322-327.

[10] J.L. Barlow and U.B. Vemulapati, A Rank Detection Method for Sparse Matrices Using a Static Data Structure, Tech. Rep. CS-90-35, Department of Computer Science, The Pennsylvania State University, University Park, PA, 1990.

[11] J.L. Barlow and U.B. Vemulapati, Convergence of Deferred Correction for Equality Constrained Least Squares Problems in Two Iterations, Tech. Rep. CS-90-36, Department of Computer Science, The Pennsylvania State University, University Park, PA, 1990.

[12] J.L. Barlow and R.J. Zaccone, Error Analysis in Unnormalized Floating Point Arithmetic, Tech. Rep. CS-90-23, Department of Computer Science, The Pennsylvania State University, University Park, PA, 1990.

[13] C.H. Bischof, D. Pierce, and J. Lewis, Incremental Condition Estimation for Sparse Matrices, to appear, SIAM J. Matrix Anal. Appl., October 1990.

[14] J.R. Bunch, C.P. Nielsen, and D.C. Sorensen, Rank-one modifications of the symmetric eigenproblem, Numerische Mathematik 31,(1978), pp.31-48.

[15] J.R. Cavallaro and I.D. Walker, Inverse Kinematic VLSI Architectures for Redundant Robots with On-Line CORDIC, Tech. Rep. 9004, Department of Electrical and Computer Engineering, Rice University, Houston, TX, 1990.

[16] T.F. Chan, G.H. Golub, and R.J. LeVeque, Algorithms for Computing the Sample Variance: Analysis and recommendations, The American Statistician 37,(1983),pp.242-247.

[17] J.W. Demmel, J.J. Dongarra, J. DuCroz, A. Greenbaum, S. Hammarling, and D. Sorensen, Prospectus for the Development of a Linear Algebra Library for High-Performance Computers, Tech. Rep. No. ANL/MCS-TM-97, Mathematics and Computer Science Division, Argonne National Laboratory, September 1987.

[18] J.W. Demmel and K. Vaselic', Jacobi's meth ' is more accurate than QR, Tech. Rep. 468, Department of Computer Science, Courant Institute of Mathematical Sciences, New York University, October 1989.

[19] J.J. Dongarra and D.C. Sorensen, A Fully Parallel Algorithm for the Symmetric Eigenproblem, SIAM J. Sci. Stat. Computing 8, (1987), pp.139-154.

[20] M.D. Ercegovac and A.L. Grnarov, On the Performance of On-line Arithmetic, Proc. of the 1980 Inter. Conf. on Parallel Processing, 1980, pp.55-61.

[21] J.A. George and E. Ng, SPARSPAK: Waterloo Sparse Matrix Package User's Guide for SPARSPAK-B, Research Report CS-84-37, Department of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, 1984.

[22] M. Gunzberger and R. Nicholaides, Elimination with noninvertible pivots, Linear Algebra and Its Applications 64, (1985), pp.183-189.

[23] S.L. Handy, On the Numerical Solution of the Eigenvalue and Inverse Eigenvalue Problem for Real, Symmetric Toeplitz Matrices, PhD Thesis, Department of Computer Science, The Pennsylvania State University, University Park, PA, May 1990.

[24] S.L. Handy and J.L. Barlow, Eigenvalue Bounds for Banded, Symmetric Toeplitz Matrices, Tech. Rep. CS-88-19, Department of Computer Science, The Pennsylvania State University, University Park, PA, 1988.

[25] S.L. Handy and J.L. Barlow, The Numerical Solution of Banded Toeplitz Eigenproblems, in preparation, 1990.

[26] J.J. More' and G. Toraldo, Algorithms for Bound Constrained Quadratic Programming Problems, Numerische Mathematik 55,(1989), pp.377-400.

[27] J.J. More' and G. Toraldo, On the Solution of Large Quadratic Programming Problems with Bound Constraints, to appear, SIAM J. Optimization 1 (1990), to appear.

[28] U.B. Vemulapati, Solving Least Squares Problems on Distributed Memory Machines, PhD Thesis, Department of Computer Science, The Pennsylvania State University, University Park, PA, December 1990.

[29] R.J. Zaccone, Numerical Properties of Digit Online Arithmetic, PhD Thesis, Deapartment of Computer Science, The Pennsylvania State University, University Park, PA, December 1984.